

Игра, которую мы записали

I hello.asm
↓ директива

global _start ← 'метка' начала вызова

section .text ← секция кода программы должна наз. так

_start: ← метка начала - т. входа в прог.

mov eax, 60 ← (вызова) системного кода exit = 60 — завершение программы

mov edi, 22 ← параметр системного вызова exit передается в edi (аналог return)

syscall ← инструкция: вызов сист. вызова что-то

II Комманды

1) В термине - перейти к месту расположения файла

2) `nasml -f elf64 hello.asm -o hello.o`
опция формат файла, в кот. хотим скомпилировать код
опция, указ. имя скомпилирован. файла

3) Объектный файл → исполняемый файл

ld -o hello hello.o

4) Запуск файла:

./hello

5) Получение кода возврата программы

echo \$? (по примеру из I это будет 22)

III Вывод строки пример

global _start

секция
данных

section .data

метка

message: db

тип
данных

строка

в ассемблере — просто набор байтов

код
символа

перевода строки

section .text

_start:

mov rax, 1 ← системного вызова функции write

mov rdi, 1 ← дескриптор стандартного вывода

mov rsi, message ← в регистр rsi загружаем адрес выводимой строки

mov rdx, 13 ← число байтов строки, кот. выведется (учитывая символ 10)

syscall

Caci. box. [mov rax, 60
exit
[sys call